Methods and tools for mining multivariate time series

De Gouveia da Costa Cachucho, R.E.

Chapter 1

Introduction

We live an era of unprecedented data challenges. These challenges have been fuelled by technological developments in storage, communication and collection of data. A good example of such developments can be recognized in sensor networks. Most people are not aware of such sources of data but they are around us, in our houses, means of transportation and even with ourselves, fitted into smartphones, watches and other wearables. These sensors have been inserted ubiquitously into our daily lives, producing large amounts of data with the potential for extensive analysis. If seen as a whole, most sensor systems produce multiple variables, derived from one or more sensors. This source of data, that we refer to as *multivariate time series*, forms the main focus of this thesis.

Mining multivariate time series is both the title and the core of this thesis. As a data mining thesis, there is a focus on the algorithmic solutions for the challenges related to multivariate time series. We envision these tasks in three different dimensions: developing methods that solve a particular mining task, providing tools that embody data mining methodologies and developing applications were temporal data plays a central role. Each dimension poses its unique challenges on how to process multivariate time series data.

1.1 Motivation

We start by introducing the main motivation behind our research in multivariate time series and the challenges that we found for the field of data mining as a field that needs to interact with other disciplines.

1.1.1 Data Science

Data-related challenges represent a major scientific effort these days. Such challenges cut across many disciplines, making the data paradigm the broadest of nowadays in terms of disciplines it can potentially affect. This context encourages the blooming of buzzwords and research hypes, such as big data and data science. These buzzwords are so strong that they boosted the focus of computer scientist towards data. As an example, over the last fifteen years, my own institute, the Leiden Institute of Advance Computer Science redirected its educational and research focus mostly towards challenges related to data, such as optimization, data mining, bioinformatics, and computer vision. More recently, terms such as big data fuelled multidisciplinary partnerships and a general understanding that not only statistics, but also computer science needs to support other scientific fields in their data analysis.

Let us start with the term *big data*. There are no strict definitions for what big data really is. This can cause considerable ontological confusion, when different disciplines get together to discuss this term and its challenges. Without pretending to have the best definition of big data, here is a (somewhat simplistic) attempt: *Big data refers to the challenges created by the fact that our present technological capacity to measure and collect data outperforms in great measure our capacity to explore and exploit it.* But then, how big is big data for most? Furthermore, does our present incapacity to deal with big data represent all the challenges around the paradigm of data?

Firstly, when talking about data, size matters. Let us start with the most extreme case, the World Wide Web. Today, if we were to collect and store all the data transferred during one day over all the World Wide Web, this would amount to a pile of burned CDs of data (roughly 700 Mb each) that would stack from Earth to Mars and back [1]. But this is not the case in the majority of data science challenges, where the scale of what is considered to be a challenging amount of data is determined by the tools available. Most of the data mining research around the term big data is about developing new computational methodologies that can upscale fairly standard analyses to Gbs or Tbs of data. Without reducing the merits of this line of research, in fact most fields of research are struggling with datasets in the order of Mbs. This struggle is in many cases due to the usage of standard tools, which are incapable of loading a large file for analysis or are simply not able to plot the data for a first inspection. Research projects need better tools, either for general purposes or in some cases tailored for a specific project.

[1] http://www.imdb.com/title/tt5275828/

So is there more to the challenges surrounding data than scalability? I believe there are, and they are multidisciplinary in nature. For a complete approach to any current data project, the scientist's background should include a good domain of databases design and management, algorithm design and performance, statistics, machine learning systems, and optimization, just to name a few. The broader field that involves this multidisciplinary knowledge has been commonly referred as *data science*.

The multidisciplinary nature of data science can be seen as a natural extension of data mining. Data mining is a field that marries methods both from mathematics such as machine learning and statistics with computer science techniques such as databases and design of efficient algorithms. The primary goal of data mining is to extract valuable information from data and turn it into a models that can be further used. Data science extends this modeling challenge both with challenges upstream and downstream.

Upstream data science challenges include turning the attention towards better data collection protocols, efficient data collection systems, faster data communication and data storage solutions (short period, long term storage, work-in-progress storage, multiple user access). For example, recently most of the principal funding agencies of Dutch research institutes (e.g. NWO, H2020 and ERC), started asking project applicants to write a data management plan for each proposal. Increasingly, domain experts in many areas of research are dealing with large amounts of generated data. Among the different research communities, one can find the social sciences, life sciences, arts and design, among others. Many of these domain experts may not have a computer science background, but often they find themselves dealing with data challenges.

Consider now the challenges downstream. They include the development of tools that include other data analysis. Such tools need to be easily available and designed for a broad audience. Additionally, there is a need to transform results of models (predictive and descriptive) into decision support systems, such that results turn into possible actions.

This thesis seeks to provide methods and methodologies for domain experts using sensor data, by building data mining algorithms tailored to sensor data. We are also developing easy-to-use standalone tools where visualization, analysis and building models of time series does not require programming skills.

The research of this thesis has been developed in multiple sensor-related projects. Referencing such projects should enforce the idea that data mining can be the common ground for multidisciplinary projects. The following sections relate to projects in the areas of civil engineering, medical sciences, life sciences (sports) and social sciences.

InfraWatch

Infrastructures such as bridges are built to endure harsh conditions, for instance as heavy traffic or extreme weather conditions. However, it is known that they do not last forever: in the long run, infrastructures deteriorate very gradually and eventually they need to be repaired or replaced. In order to keep traffic moving, bridge owners need to do maintenance. How to predict the necessity for maintenance? The current approach is to do *ad-hoc* external inspections.

InfraWatch is a project that joins asset managers, contractors and academia (both civil engineering and computer sciences) to change the current practice of bridge maintenance. A large sensor network (approximately 145 sensors) was installed on a highway bridge near Amsterdam, collecting data since 2008 at a sampling rate of 100 Hz, generating big data. At the Leiden Institute of Advanced Computer Science, we have been developing algorithms that build data-driven models of the bridge [102, 103, 67, 15] and new visualization tools [101, 65, 13], for large multivariate time series.

Leiden Longevity Study

The baby boomers of the fifties and sixties are aging and as a consequence in Europe, health has become a major concern for people themselves [96], policy makers [92] and science [104, 94]. The *Leiden Longevity Study* is designed to identify genetic and phenotype markers that relate to longevity. A total of 421 families were recruited, consisting of long-lived white siblings, their offspring and the offspring's partners (who act as controls). The collection of data was vast and for some cohort studies, the data collection protocol included unsupervized sensor data (accelerometers and bioharness) for long periods of free living. The question became then: how to extract valuable information about physical activity that can help explaining the aging process.

In order to explore the unsupervized data, first a validation study (GOTOv) produced a dataset with a set of activities that have been annotated (hence a supervised dataset). The participants of this study have the same characteristics as the population as the Leiden Longevity Study studies. Using a method described in this thesis [14], we have been able to create tools (see Chapter 5) and highly reliable activity recognition models [74]. This will allow us to explore the data from the cohorts that have unsupervized data.

Social Competence

Social interactions in the playground have been considered important learning opportunities, for children to learn social skills at preschool years. Specifically, all forms of social play (fantasy, role, exercise, or rough-and-tumble) have been related to children's social competence [21]. The research team Focus on Emotions, of the Faculty of Social and Behavioural Sciences of Leiden University and the Human Motion Faculty of the University of Lisbon have been studying child's play in the playground.

As data science partners, we designed a new data collection protocol to measure interactions in the playground, based on Radio-Frequency Identification Devices. Using active RFIDs as badges, the interactions could be measured for all the children at the same time at a fairly high sampling rate (4 Hz). This partnership resulted in several new analyses [99, 100], as well as a first place in a smart city competition [1]. Collection of additional datasets is ongoing.

Sports Analytics: MASS project

The MASS project focuses on the monitoring and analysis of elite Dutch speed skaters during several training seasons. The goal is to consider a wide range of aspects, in order to find effective patterns in their training routines, which might lead to more effective training and better competition performance. The project builds on a unique collection of daily data about Dutch speed skaters. Collected over the last ten years, this data comes from within LottoNL-Jumbo, the top-level team of coach Jac Orie [73].

In order to mine this data, first we set up a database infrastructure to store the data and developed a web-based tool where one can make data and model analysis on the fly with ease. The research resulted in a high profile article [53] (see Chapter 6) as well as considerable publicity.

1.1.2 Research Questions

The applied projects outlined above, as well as many similar projects producing complex, multivariate time series, offer many opportunities for new research in the area of data science. The following is a list of research questions inspired by these projects that motivate the work outlined in this thesis. The first two questions have to do with the multivariate nature of the data itself:

> **Q1** How can we find unsupervised patterns in multivariate time series?
>
> **Q2** Can we find dependencies between variables in multivariate data that occur intermittently?

These two questions relate to applications where the observed system can be in different states with different observed behaviour, but it is not quite clear yet which are the states it can be in. In other words, this is a so-called unsupervised setting, where one would like the data to be clustered into a finite number of states, but the states should be reasonably steady, and not switch from one time point to the next. This setting is for example relevant in the Leiden Longevity Study, where one would like to recognise various activities in a person's behaviour. Additionally, research question Q2 relates to the possibility that different states show different phenomena with respect to the measured variables involved. Research questions Q1 and Q2 are addressed in Chapter 2.

The unsupervised setting is attractive if one is unfamiliar with the dataset and domain it captures. However, in many cases, one is dealing with a supervised task, and either regression or classification models need to be induced from multivariate data. A specific challenge, that was also present in some of the projects mentioned above, is that the supervised 'labels' are not always available at the same rate as the detailed multivariate data was recorded. For example, it is easy to capture activity data at high sampling rates, e.g. by means of an accelerometer, but obtaining the activity of a person can realistically not be recorded more often than once per second, if not much slower. This inspires the following research questions, which are addressed in Chapters 4, 5 and 6:

> **Q3** How can we learn classification and regression models from multivariate time series with mixed sampling rates, taking into account the different temporal scales at which dependencies might hold?

> **Q4** How can we automatically derive good time series features that take into account delays and temporal aggregation?

In the field of sports analytics, one is interested in reliable models, but also in finding patterns that are actionable and interpretable for a coach, in order to optimise the effectiveness of their training routines. The MASS project above motivated research question Q5:

> **Q5** How can interpretable and simple patterns be extracted from multivariate training data, while considering the complexities of elite sports and of how the human body responds to various training impulses?

The above research questions relate to key data science methods that are required to deal with the data challenges described, but they over-simplify the process of data science. In practice, a project requires a methodology and tools to support the analysis of complex data. The following research question addresses this issue, which is also the topic of Chapters 3, 5, and parts of Chapter 6.

> **Q6** How can the non-trivial analysis of multivariate times series by supported by a methodology and a tool?

1.2 Multivariate Time Series

In this section, we introduce the fundamental concepts used throughout this thesis. To start, consider a univariate time series, defined as follows:
Definition 1. *A* univariate time series $\mathbf{t}$ *is a finite sequence of values* $(t_1, \ldots, t_n)$ *ordered in time, where* i *represents an index of the sequence* $i \in \{1, \ldots, n\}$ *and* t_i *is a real value:* $\forall i \ \ t_i \in \mathbb{R}$.

Consider now the properties of a single time series, $\mathbf{t}$. In this univariate setting, the sequence of data points $t_i, i \in \{1, \ldots, n\}$ represents a process measured over a period of time.

Across all the projects mentioned above and chapters following this introduction, there is a data structure that is common to all: the *multivariate* time series. Continue to assume that our time series is of finite length n. The extension to the definition above is that now we have m variables to consider. We formalize a multivariate time series as a matrix $\mathbf{T}$ of size $n \times m$, as follows:

$$\mathbf{T} = \begin{pmatrix} \mathbf{T}_{1,1} & \mathbf{T}_{1,2} & \cdots & \mathbf{T}_{1,m} \\ \mathbf{T}_{2,1} & \mathbf{T}_{2,2} & \cdots & \mathbf{T}_{2,m} \\ \vdots & \vdots & \mathbf{T}_{i,j} & \vdots \\ \mathbf{T}_{n,1} & \mathbf{T}_{n,2} & \cdots & \mathbf{T}_{n,2} \end{pmatrix}$$

Please note that in this matrix, there is a temporal order: $\mathbf{T}_{i,j}$ represents a measurement at time point i for variable j. The temporal order of the data can be represented as a sequence $\{1, 2, \ldots, i, \ldots, n\}$. Note that this representation assumes that all measurements for the different variables j are performed at the same time.

1.2.1 Sampling Rates

Definition 1 is an abstract formalization of a time series without specifications of the exact timing of the data. It is abstract because the series is just ordered by an index from $\{1, \ldots, n\}$. In reality, however, each row in the series is actually measured at a specific moment in time, which we refer to as the *timestamps* of the measurement. A timestamp can be seen as a function $\omega(i)$ that maps the index i to a specific point in time.

Having the timestamps, one important property to know is which is the temporal spacing between consecutive measurements. For consecutive measurements, this spacing is simply:

$$\omega(i) - \omega(i-1), \forall i \in \{2, \ldots, n\}.$$

A more informative measure of how measurements are spaced is actually the *sampling rate*, which is the average number of samples per unit of time:

$$f = \frac{n-1}{\omega(n) - \omega(1)}.$$

The sampling rate is a frequency with unit s^{-1} (per second), more generally referred to as Herz (Hz). The sampling rate f is an average statistic that holds for the entire series. In contrast, the sampling rate can also be computed locally, as follows:

$$f_i = \frac{1}{\omega(i) - \omega(i-1)}, \forall i \in \{2, \ldots, n\}.$$

Fixed Sampling Rate The most desirable situation is when the sampling rate is fixed, so the temporal spacing between measurements is always the same. Note that well-behaved periodicity of the data will allow us to ignore time stamps and simply use indexes. We refer to *fixed sampling rate* when the sampling rate f_i is always constant, such that

$$f = f_i = \frac{1}{\omega(i) - \omega(i-1)}, \forall i \in \{2, \ldots, n\}.$$

Variable Sampling Rate When the statement above does not hold, we refer to it as *variable sampling rate.* Notice that variable sampling rate is often the case, sometimes due to imprecision in measuring, other times because the measured phenomena does not have a constant periodicity.

There are multiple standard approaches to deal with variable sampling rates. If the differences between sampling rates f_i is very small, then one can just opt to ignore it. Alternatively, one could find solutions either from the data perspective or from the design of the algorithm. From the perspective of the data, one can have a pre-processing step to make the sampling rate constant, by using standard techniques such as interpolation or repeating values [105, 83]. From the algorithmic perspective, one can design and implement solutions to deal with variable sampling rates, with techniques such as buffering [9].

1.2.2 Machine Learning Tasks

The machine learning literature identifies a range of standard tasks. These differ in terms of nature of the data, such as the traditional tabular data, as opposed to structured data (e.g. time series). Machine learning tasks can also differ in terms of aim, such as classification, regression, biclustering, subgroup discovery and segmentation. This suggests a landscape of problems and solutions where it is easy to get lost.

One good way to categorize machine learning tasks is to divide them into two learning paradigms: unsupervized and supervized. This nomenclature separates tasks with a clear, known and measurable target (supervized) and tasks that try to make sense of data without having a prior target (unsupervized). This thesis focuses on tasks from both paradigms that are relevant for multivariate time series, as mentioned bellow.

Unsupervized Learning

The field of unsupervized learning is vast, and reflects the different ways one could extract meaningful information from data. As for time series, examples of well-known machine learning tasks are *whole time series clustering* [22, 49, 2], *segmentation* [35, 46, 47] and *motif discovery* [72, 20, 71, 103, 88, 70, 93].

Whole time series clustering applies when one is facing a dataset where the same variable was measured in multiple systems, under the same protocol. Take as an example measuring the core body temperature of a person. If this would be done for multiple persons, then we could cluster the different types of metabolism in terms of temperature [22]. Although common, this setting does not apply to our definition of multivariate time series, where it is assumed that we are measuring only one system.

Let us now focus on the situation of measuring one system over time. In such situations, the most well-known tasks are *segmentation* and *motif discovery.* On the one side, segmentation aims to cut a time series into a set of segments, where the data inside each segment is homogeneous. On the other side, motif discovery finds recurrent patterns or motifs in the time series, allowing the possibility of selecting only some parts of the time series. Both segmentation and motif discovery are tasks that typically are applied to univariate data, not easily to multivariate time series.

In this thesis, we introduce an unsupervized task that is applicable to multivariate data collected from a single system: biclustering of multivariate time series. Biclustering is a well-known task in the machine learning literature, but cannot be readily applied to time series data. We reinterpret this task into a scenario of multivariate time series. An introduction to this task, a formal definition, an algorithmic solution and tool can be found in Chapters 2 and 3.

Supervized Learning

Consider now that we have a known and measurable target to model. In such a case, we are dealing with a different paradigm from the one above, known as supervized learning. In such a case, one wants to predict a target. In the case of this thesis, we will focus on predictions in a multivariate scenario.

Before moving on to the definition of the target, let us make an important distinction between prediction of future data (forecast a system) and prediction of current and past data (understand a system). Forecasting is to estimate future values given past values of a time series. Please note that forecasting can be done both in univariate and multivariate settings. Unlike forecasting, prediction of existing data seeks ways to explain some target in terms of other variables. This sort of prediction can be divided into regression or classification problems. Such models are invariably multivariate. The fundamental difference to forecast resides in the error. The regression or classification error is the difference between the actual value of the estimation and the prediction of such value. We focus on this set of tasks and formalize them bellow.

Consider that besides the m independent variables present in $\mathbf{T}$, we have a dependent variable $\mathbf{r}$. Having $\mathbf{r}$ as a target, the general task in this situation is to find a model $\mathcal{F}$, such that:

$$\mathbf{r}_i = \mathcal{F}(\mathbf{T}_i) + e_i$$

Please note that e_i represents the error of the model for the prediction of existing data as mentioned a couple of paragraphs above. For both classification and regression models, the objective is to minimize this error.

Classification A problem can be categorized as a classification problem when the dependent variable $\mathbf{r}$ is categorical. When categorical, the domain of $\mathbf{r}$ assumes a finite number c of possible, such that $\mathbf{r}_i \in \{1, \ldots, c\}$. To evaluate a classification model, normally one would use accuracy as a metric. To define the accuracy, please consider a binary classification problem ($c = 2$), with the following terminology in terms of prediction.

$$accuracy = \frac{number\ of\ cases\ correctly\ predicted}{number\ of\ cases}$$

Please note that this accuracy rate can be generalized to a multiclass classification problem where $c > 2$. The accuracy reflects the quality of a classification model, due to its capability to correctly predict the classes. The closer to unity, the better are the predictions performed by the classifier.

Regression Next to classification problems, we have regression problems. A regression problem is recognized as such when the dependent variable $\mathbf{r}$ is numeric, such that $\mathbf{r}_i \in \mathbb{R}$. In such a numeric setting, the error can be either an overestimation or an underestimation and is calculated as follows:

$$e_i = \sqrt{(\mathcal{F}(\mathbf{T}_i) - \mathbf{r}_i)^2}$$

Please note that as measure of quality for a regression model, one would want a normalized measure to be able to compare methods across different datasets or just to avoid the interpretation using the domain size of the dependent variable $\mathbf{r}$. For such a measure, we normally adopt r-squared (R^2), which is defined as follows:

$$R^2 = \frac{\sum_{i=1}^{n} {e_i}^2}{\sum_{i=1}^{n} (\mathbf{r}_i - \bar{\mathbf{r}})^2},$$

where $\bar{\mathbf{r}}$ is the average of the dependent variable. R-squared is a measure of quality of the regression model by comparison with a simple average estimation. The closer to unity, the best is the model and the further is from being an average estimation.

Mixed sampling rates In many cases, the sampling rate for $\mathbf{T}$ is very high or at least higher than desired as an output. Here as output, we refer to the result of a regression or classification task as defined above. As an example, please consider the InfraWatch bridge mentioned previously, with an installed sensor network that measures at 100 Hz and where at best, we want to know the traffic intensity every minute. For such situations, we aim to deal with modeling tasks where the sampling rate of the dependent variable $f_{\mathbf{r}}$ is (much) lower than that of the independent variables $\mathbf{T}$

$$f_{\mathbf{r}} \leq f_{\mathbf{T}}.$$

Please note that a consequence of such an assumption is that the length of $\mathbf{r}$ is not n anymore but $|\mathbf{r}|$. The differences of sampling rates imply differences in the number of samples between $\mathbf{T}$ and $\mathbf{r}$, where $|\mathbf{r}| < n$. The regression or classification models as stated above won't hold under such circumstance and one would need to change the data such that for each value of the dependent variable $\mathbf{r}$, there is only one vector of values as independent variables.

The transformation of independent variables we normally call feature construction. As described above, in the supervized settings we are looking for a function $\mathcal{F}$ that is able to explain $\mathbf{r}$. In the case of mixed sampling rates, we have a situation that $\mathbf{T}$ cannot be used to model $\mathbf{r}$ directly. To model $\mathbf{r}$, we need to transform $\mathbf{T}$ into a feature set $\mathbf{F}$, such that for any of the features that belong to the set $\mathbf{F}$ have length $|\mathbf{r}|$. We approach this transformation of $\mathbf{T}$ by means of *aggregation*, which is described below.

1.2.3 Aggregation of Time Series

In this section, we discuss the relevance of so-called *aggregation functions* to deal with time series. Starting from the beginning, one could ask what an aggregation function does. In short, it takes a vector of numbers and transforms it into a lower dimension, typically a single value. As an example, let's consider a univariate time series $\mathbf{t}$ according to definition 1. An aggregation function summarizes $\mathbf{t}$ as follows:

$$a : \mathbb{R}^n \rightarrow \mathbb{R}^1.$$

Please note that the usage of an aggregation function such as a, intends not only to perform a dimensionality reduction but also to summarize some particular informative aspect of $\mathbf{t}$.

There are many ways to summarize sequences of measurements. To consider such variety, we assume a set of aggregation functions A, where $a \in A$. Although the set A can theoretically be defined to include an infinite number of functions, we will restrict A to a very manageable number. We consider A to be a family of well-known aggregation functions, normally designated as *descriptive statistics*, such as a minimum, a maximum or an average. More on A can be found in Chapter 4.

Let's now focus on how to properly use aggregation functions in the context of time series. Above, we considered applying such functions to summarize the whole time series $\mathbf{t}$. Although it works well as an example, this is not how aggregation is applied in most practical cases. More commonly, a is applied to subsequences of a time series, informally referred to as a *window*. A time series subsequence is defined as follows:

Definition 2. *Given a univariate time series* $\mathbf{t}$ *as in Definition 1, a subsequence* $\mathbf{s}_{\mathbf{t},i,l}$ *is a subset of* $1, \ldots, n$*, containing* l *contiguous values. We consider* $\mathbf{s}_{\mathbf{t},i,l}$ *to be represented by a vector* $(t_{i-l+1}, \ldots, t_{i-1}, t_i)$*, and:*

- $i \in \{l, \ldots, n\}$,
- $\forall i \;\; t_i \in \mathbb{R}$,
- $1 \leq l \leq n$.

Aggregation functions, when applied to time series subsequences, will summarize a particular period of time. Such usage of aggregation function in subsequences of time series is normally intended for feature construction. In the context of this thesis, the need for feature construction has already been motivated above, while discussing how to mine multivariate time series with mixed sampling rates.

One could now consider how a feature is constructed in the context of a time series. For the purpose of simplicity, let us start by considering a feature that has the same sampling rate as the original time series. This implies that the aggregation function slides orderly through the subsequence (one data value in, one out). A feature can then be constructed as follows:

$$f_{t,a,l}[i] = a(\mathbf{s}_{\mathbf{t},i,l})$$

A more generic definition of time series feature (aggregate feature) is given in Chapter 4. Please note that we assumed for now that features are constructed with subsequences of the same length l. Such an assumption works fine when we have a fixed sampling rate. As presented in Section 1.2.1, it can also occur that one faces data with variable sampling rates. In such a scenario, there are two solutions to consider:

- If the variation of the sampling rate is considerable, one could fix a time interval and include all the data that falls into that window of time.

- If the variation of the sampling rate is insignificant, one could just do some pre-processing to adjust the sampling rate and still fix the size of the window based on the index.

A feature as defined above captures some particular aspect of a time series. Following this idea, we identify the following applications of aggregate features:

- Feature engineering: capture or extract some specific aspect of the time series that can potentially be informative, and is not apparent in the original time series.

- Smoothing: transform the time series into a smoother version of itself. Here the idea is to reduce the noise that naturally occurs in data.

- Integration over time: with integration over time we mean extracting some aspect of the data that exhibits an influence over some time.

- Delays: the idea of capturing delays is important in time series because sometimes the causal effect is not instantaneous. Being able to capture information with delays might help to do such integration over time.

1.3 Main Contributions

In this section, we describe the main contributions of this thesis. The contributions will be organized by chapters that follow this introduction. Each chapter corresponds to one article. Three of the chapters have been published in the proceedings of computer sciences conferences, one in a high impact journal and a remaining one that is under submission process.

biclusTS: Biclustering Multivariate Time Series In Chapter 2, we introduce biclustering as a relevant task for multivariate time series data. Although biclustering is well-established as a machine learning task, it is primarily considered in the context of classical tabular data [19, 61, 79]. Existing solutions do not work with multivariate time series because such algorithms may pick single rows (in this context, timestamps). When applied to multivariate time series, a classical algorithm will produce biclusters with cluster members scattered across all the time series, with little or no relevance when seen from a temporal perspective.

Our algorithm uses as coherence measure between segments, namely a one-step estimation of the density difference between distributions of values within different segments. Such an estimation of the density difference differs from the traditional two-step approach that first estimates the probability density functions and then calculates the estimated density difference of the two distributions. The work in this chapter was published as follows:

> Ricardo Cachucho, Siegfried Nijssen, Arno Knobbe, *Biclustering Multivariate Time Series*, in Proceedings of Intelligent Data Analysis (IDA), 2017

Bipeline: Bisclustering tool In Chapter 3, we introduce a working tool, easily accessible to everyone that wants to experiment with biclustering methods in multivariate time series. We called this tool *Bipeline*, due to the main idea of allowing users to follow easily an experimental pipeline. Such a pipeline means that this tool can be used to load, visualize, preprocess and bicluster multivariate time series. In terms of biclustering, the intention of Bipeline is not only to share the algorithm biclusTS (Chapter 2), but also to compare it to other standard biclustering algorithms. The following paper describes the tool own detail:

> Ricardo Cachucho, Kaihua Liu, Siegfried Nijssen, Arno Knobbe, *Bipeline: a Web-based Visualization Tool for Biclustering of Multivariate Time Series*, in Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD), 2016

Accordion: Mining Mixed Sampling Rates In Chapter 4, we change the learning paradigm to a supervized scenario. Here, we introduce a new method that deals with multivariate time series with *mixed sampling rates*. We named this algorithm *Accordion*. Accordion automatically searches for good aggregate features for a given dataset with mixed sampling rates. As a result, Accordion returns a set of aggregate features at the sampling rate of the dependent variable.

What sets our method apart from many has to do with the way good descriptive features are searched automatically. While the majority of classification and regression algorithms see the feature construction as an independent preprocessing step, we see it as a machine learning problem. The construction of good aggregate features is an optimization procedure, and as a result, good features will lead to good models. The obvious question is then, what is a good feature? We consider a feature to be good, when it is able to properly discriminate target classes (classification) or when it has a good correlation with a numeric target (regression). This work was publishes as:

> Ricardo Cachucho, Marvin Meeng, Ugo Vespier, Siegfried Nijssen, Arno Knobbe, *Mining Multivariate Time Series with Mixed Sampling Rates*, in Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp), 2014

ClaRe: Classification and Regression tool In Chapter 5, we introduce a web-based tool (ClaRe) to build regression and classification models using features from the Accordion algorithm (Chapter 4).

Our tool aims to make Accordion accessible in a SaaS (Software as a Service) environment. Given a multivariate time series dataset, it allows users to easily access Accordion's features. The upside of such a SaaS is the ability to access Accordion embedded in a data mining methodology: load, pre-process, visualize and evaluate. The following paper is under submission:

> Ricardo Cachucho, Stelios Paraschiakos, Kaihua Liu, Benjamin van der Burgh, Arno Knobbe, *ClaRe: Classification and Regression Tool for Multivariate Time Series*, in Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD), 2018

Speed Skating Analytics Chapter 6 reports on a cooperation between an elite speed skating team (LottoNL-Jumbo) and Leiden University, where the objective is to analyze and optimize the speed skating performance of individual athletes. To look into performance, we accessed daily training and competition results data, collected over a period of over 15 years.

From a modeling perspective, the challenge was to find meaningful aggregate features to avoid situations of under- or overtraining in the athletes' training schedules. The challenge from the data science perspective was on how to weigh such aggregate periods and on how to deal with variable sampling rates in multivariate time series. The models presented represent a mixture of linear models and subgroup discovery (step functions) that should provide a platform of analysis for personalized performance optimization. The content of this chapter appears in the following publication:

> Arno Knobbe, Jac Orie, Nico Hofman, Benjamin van der Burgh, Ricardo Cachucho, *Sports Analytics for Professional Speed Skating*, in journal of the Data Mining and Knowledge Discovery, Volume 31, Issue 6, pp 1872–1902, Springer, 2017

Please note that, in the the case of this publication I am not the first author. In this case, I designed and developed the application where the experiments ran, conducted the experiments to find meaningful aggregate features used in the performance models and lead LIACS data scientist.

1.4 Thesis outline

Following this introductory chapter, this thesis presents a series of papers. These are papers that have been published and peer reviewed, with the exception of Chapter 5 that has been submitted for publication. The papers are presented in the form of self-contained chapters. Although being self-contained chapters, one could divide the work into chapters on unsupervized tasks and chapters on supervized tasks.

Unsupervized learning Chapter 2, Biclustering Multivariate Time Series [15], presents a method on how to select subsets of rows (time periods) and subsets of columns (variables) under a coherence measure. This paper was published at the Intelligent Data Analysis 2017 conference.

Chapter 3, Bipeline: a Web-based Visualization Tool for Biclustering of Multivariate Time Series [13], presents a web-based tool where users can test different biclustering algorithms in multivariate time series. This paper was published at the ECML-PKDD 2016 conference.

Supervized learning In Chapter 4, we change to a supervized learning setting. This chapter, Mining Multivariate Time Series with Mixed Sampling Rates [14], presents a method to search for good aggregate features to build decision trees and linear regression models. This paper has been submitted and presented at the Ubiquitous Computing 2014 conference.

Chapter 5, ClaRe: Classification and Regression Tool for Multivariate Time Series, presents a web-based tool where the users can experiment with aggregate features as presented in Chapter 4 within a data mining methodology framework. This paper was published at the ECML-PKDD 2016 conference.

As for Chapter 6, Sports Analytics for Professional Speed Skating [53], we present an empirical study on how to apply data science techniques to a professional sports environment. This paper was published in the Data Mining and Knowledge Discovery journal.

Finally, in Chapter 7 we give an overview of the findings and contributions that this thesis proposes.